

## Дводимензионални низи во C++

Повеќедимензионалните низи претставуваат низи од низи. Сите елементи од една повеќедимензионална низа мора да бидат од ист тип - важи истото ограничување што важеше и кај еднодимензионалните низи.

Да разгледаме една матрица (дводимензионална низа) која содржи елементи од податочниот тип `int`:

```
int mat[5][4]
```

<code>mat[0][0]</code>	<code>mat[0][1]</code>	<code>mat[0][2]</code>	<code>mat[0][3]</code>
<code>mat[1][0]</code>	<code>mat[1][1]</code>	<code>mat[1][2]</code>	<code>mat[1][3]</code>
<code>mat[2][0]</code>	<code>mat[2][1]</code>	<code>mat[2][2]</code>	<code>mat[2][3]</code>
<code>mat[3][0]</code>	<code>mat[3][1]</code>	<code>mat[3][2]</code>	<code>mat[3][3]</code>
<code>mat[4][0]</code>	<code>mat[4][1]</code>	<code>mat[4][2]</code>	<code>mat[4][3]</code>

Дводимензионалните низи, во C++, се декларираат со следната наредба:

```
tip ime[R][C];
```

Притоа, `tip` го означува податочниот тип на секој од елементите во дводимензионалната низа, `ime` го означува името на променливата, додека `R` и `C` го означуваат бројот на редови и колони, соодветно. На пример, матрицата од сликата дадена погоре може да се декларира со наредбата `"int mat[5][4];"`

Индексите на елементите во дводимензионалните низи се движат од 0 до `R-1` (за редовите) и од 0 до `C-1` (за колоните) - слично како и кај еднодимензионалните низи. Пристапот до елементите се изведува преку наведување на името на променливата (со која е декларирана дводимензионалната низа) и индексот на редот и колоната, наведени едно по друго (`mat[0][0]`, `mat[0][1]`, `mat[0][2]`, ..., `mat [1][0]`, `mat [1][1]`, ..., `mat [2][0]`, `mat [2][1]`, итн).

Дводимензионална низа може да се декларира и на следниов начин:

```
int A[4][3] = {{22, 23, 10},  
              {15, 25, 13},  
              {20, 74, 67},  
              {11, 18, 14}};
```

ако нема доволно елементи, останатите се 0.

Пр.1. програма со која се внесуваат и се печатат елементите на матрица

```
#include <iostream>  
using namespace std;  
int main()  
{  
    float a[50][50];  
    int m,n,i,j;  
    cout<<"Kolku redici ima matricata?"<<endl;  
    cin>>m;  
    cout<<"Kolku koloni ima matricata?"<<endl;  
    cin>>n;  
    cout << "Vnesi gi elementite na matricata: " << endl;  
    for(i=0;i<m;i++)  
    {  
        for(j=0;j<n;j++)
```

```

    {
    cin>>a[i][j];
    }
    cout<<endl;
}
cout<<"\nVnesena e matricata:"<<endl;
for(i=0;i<m;i++)
{
    for(j=0;j<n;j++)
        cout<<a[i][j]<<"\t";
    cout<<endl;
}
return 0;
}

```

Пр.2. програма со која се пресметува збир на елементите на матрица со м редици и н колони

```

#include <iostream>
using namespace std;
int main()
{
    int a[50][50];
    int m,n,i,j,s;
    cout<<"Kolku redici ima matricata?"<<endl;
    cin>>m;
    cout<<"Kolku koloni ima matricata?"<<endl;
    cin>>n;
    cout << "Vnesi gi elementite na matricata: " << endl;
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            cin>>a[i][j];
        }
        cout<<endl;
    }
    s=0;

    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
            s+=a[i][j];
    }
    cout<<"Zbirot na elementite e "<<s;
return 0;
}

```

Пр. Да се внесе матрица со димензии  $m \times n$ , потоа да се преброи колку има парни и непарни елементи во матрицата.

```

#include <iostream>

```

```

using namespace std;
int main()
{
    int a[50][50];
    int m,n,i,j,s1,s2;
    cout<<"Kolku redici ima matricata?"<<endl;
    cin>>m;
    cout<<"Kolku koloni ima matricata?"<<endl;
    cin>>n;
    cout << "Vnesi gi elementite na matricata: " << endl;
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            cin>>a[i][j];
        }
    }
    s1=0;
    s2=0;
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            if(a[i][j]%2==0) s1+=a[i][j];
            else s2+=a[i][j];
        }
    }
    cout<<"Zbirot na parnite elementi e "<<s1<<endl;
    cout<<"Zbirot na neparnite elementi e "<<s2;
return 0;
}

```

Пр. програма со која се формира нова матрица с чии елементи се збир на соодветните елементи на две матрици а и б со м редици и н колони.

```

#include <iostream>
using namespace std;
int main()
{
    int a[50][50],b[50][50],c[50][50];
    int m,n,i,j,s,s2;
    cout<<"Kolku redici ima matricata?"<<endl;
    cin>>m;
    cout<<"Kolku koloni ima matricata?"<<endl;
    cin>>n;
    cout << "Vnesi gi elementite na matricata a: " << endl;
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            cin>>a[i][j];

```

```

    }
}
cout << "Vnesi gi elementite na matricata b: " << endl;
for(i=0;i<m;i++)
{
    for(j=0;j<n;j++)
    {
        cin>>b[i][j];
    }
}

for(i=0;i<m;i++)
{
    for(j=0;j<n;j++)
    {
        s=0;
        s=a[i][j]+b[i][j];
        c[i][j]=s;
    }
}
cout<<"\n novata matrica e "<<endl;
for(i=0;i<m;i++)
{
    for(j=0;j<n;j++)
    {
        cout<<c[i][j]<<"\t";
    }
    cout<<endl;
}
return 0;
}

```

Пр. програма со која се пресметува производ на елементите поголеми од 20 во матрица а со м редици и н колони.

```

#include <iostream>
using namespace std;
int main()
{
    int a[50][50];
    int m,n,i,j,p=1;
    cout<<"Kolku redici ima matricata?"<<endl;
    cin>>m;
    cout<<"Kolku koloni ima matricata?"<<endl;
    cin>>n;
    cout << "Vnesi gi elementite na matricata a: " << endl;
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            cin>>a[i][j];

```

```

    }
}
for(i=0;i<m;i++)
{
    for(j=0;j<n;j++)
    {
        if(a[i][j]>20) p*=a[i][j];
    }
}
cout<<"proizvodot na elementite pogolemi od 2 e "<<p;

return 0;
}

```

Пр. програма со која се пресметува збир на елементите на квадратна матрица со ред n.

```

#include <iostream>
using namespace std;
int main()
{
    int a[100][100],n,i,j,s;
    cout << "vnesi go n"<<endl;
    cin>>n;
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
            cin>>a[i][j];
    }
    s=0;
    for(i=0;i<n;i++)
        s+=a[i][i];
    cout<<"Zbirot na elementite e "<<s;
    return 0;
}

```

Пр. програма со која се пресметува колку елементи на главната дијагонала се позитивни.

```

#include <iostream>
using namespace std;
int main()
{
    int a[100][100],n,i,j,s;
    cout << "vnesi go n"<<endl;
    cin>>n;
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
            cin>>a[i][j];
    }
    s=0;
    for(i=0;i<n;i++)
    {

```

```

    if(a[i][i]>0) s++;
}
cout<<"brojot na pozitivni elementi e "<<s;
return 0;
}

```

Пр. програма со која се внесува и печати матрица  $m \times n$ , а потоа да се печати транспонираната матрица (транспонирана матрица, редиците се менуваат со колоните и обратно)

```

#include <iostream>
using namespace std;
int main()
{
    int a[100][100],b[100][100],m,n,i,j,s;
    cout << "vnesi go m"<<endl;
    cin>>m;
    cout << "vnesi go n"<<endl;
    cin>>n;
    cout << "vnesi gi elementite na matricata "<<endl;
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
            cin>>a[i][j];
    }
    cout << "vnesena e matricata "<<endl;
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
            cout<<a[i][j]<<"\t";
        cout<<endl;
    }
    cout<<endl;
    for(i=0;i<n;i++)
    {
        for(j=0;j<m;j++)
        {
            b[i][j]=a[j][i];
        }
    }
    cout << "transponiranata matrica e "<<endl;
    for(i=0;i<n;i++)
    {
        for(j=0;j<m;j++)
            cout<<b[i][j]<<"\t";
        cout<<endl;
    }
    return 0;
}

```

Пр. програма со која се внесува и печати матрица  $m \times n$ , а потоа да се испечатат елементите на споредната дијагонала.

```
#include <iostream>
using namespace std;
int main()
{
    int a[100][100],n,i,j;

    cout << "vnesi go n"<<endl;

    cin>>n;

    cout << "vnesi gi elementite na matricata "<<endl;
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
            cin>>a[i][j];
    }
    cout << "vnesena e matricata "<<endl;
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
            cout<<a[i][j]<<"\t";
        cout<<endl;
    }
    cout<<endl;

    cout << "elementi na sporedna dijagnala se "<<endl;
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
            if(i+j==n-1) cout<<a[i][j]<<" ";
    }
    return 0;
}
```

Пр. програма со која се печати максималниот елемент од секој ред од матрица со  $m$  редици и  $n$  колони.

```
#include <iostream>
using namespace std;
int main()
{
    int a[50][50],m,n,i,j,max;
    cout<<"Vnesi m"<<endl;
    cin>>m;
    cout<<"Vnesi n"<<endl;
    cin>>n;
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
```

```

    {
        cin>>a[i][j];
    }
}
for(i=0;i<m;i++)
{
    max=a[i][0];
    for(j=0;j<n;j++)
    {
        if(a[i][j]>max) max=a[i][j];
    }
}
cout<<"Najgolem e "<<max;
return 0;
}

```

Пр. Да се внесе матрица со димензии  $m \times n$ , потоа за секоја колона да се отпечати бројот на парните елементи.

```

#include <iostream>
using namespace std;
int main()
{
    int a[50][50],m,n,i,j,bp;
    cout<<"Vnesi m"<<endl;
    cin>>m;
    cout<<"Vnesi n"<<endl;
    cin>>n;
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            cin>>a[i][j];
        }
    }

    for(j=0;j<n;j++)
    {
        bp=0;
        for(i=0;i<m;i++)
        {
            if(a[i][j]%2==0) bp++;
        }
        cout<<"Vo "<<j<<" kolona"<<" ima "<<bp<<" parni elementi"<<endl;
    }

    return 0;
}

```

Пр. Да се внесе и отпечати матрица со димензии  $m \times n$ , потоа за секој ред да се отпечати збирот на парните елементи.



Пр. Да се внесе и отпечати квадратна матрица со димензии  $n \times n$ , потоа да се соберат позитивните елементи на главната дијагонала.

Пр. Да се внесе и отпечати квадратна матрица со димензии  $n \times n$ , потоа да се формира и отпечати низа од елементите на споредната дијагонала

```
#include <iostream>
using namespace std;
int main()
{
    int a[50][50],b[50],n,i,j,k=0;
    cout<<"Vnesi n"<<endl;
    cin>>n;
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
        {
            cin>>a[i][j];
        }
    }

    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
        {
            if(i+j==n-1)
            {
                b[i]=a[i][j];
            }
        }
    }
    cout<<"Elementi na sporednata dijagonala se"<<endl;
    for(i=0;i<n;i++)
        cout<<b[i]<<" ";
    return 0;
}
```